TRƯỜNG ĐẠI HỌC SƯ PHẠM TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN

PROPOSAL

QR CODE DETECTION

Nhóm sinh viên thực hiện:

- Nguyễn A Đam 4501104044
- Nguyễn Văn Phong 4501104175

Giáo viên hướng dẫn: TS. Ngô Quốc Việt

TP.HCM, Tháng 12-2021

LỜI CẢM ƠN

Với những lời đầu tiên, chúng em xin dành lời cảm ơn đến thầy TS. Ngô Quốc Việt và Quý Thầy Cô Trường Đại học Sư Phạm Thành phố Hồ Chí Minh đã tận tình truyền dạy kiến thức trong quá trình học tập tại trường. Những kiến thức đã giúp đỡ rất nhiều trong việc học tập và nghiên cứu của em.

Em cũng xin gửi lời cảm ơn đến gia đình cũng như những người thân, bạn bè đã giúp đỡ và tạo điều kiện tốt trong quá trình học tập cũng như nghiên cứu.

Trân trọng

Thành phố Hồ Chí Minh, ngày 18 tháng 12 năm 2021

MỤC LỤC

I. TÍNH CẤP THIẾT

QR Code đã trở nên quá phổ biến trong đời sống: các dịch vụ thanh toán, mạng xã hội, thông tin sản phẩm, trao đổi thông tin,…

II. MỤC TIÊU

- Xác định được vị trí **QR Code** và đưa về đúng vị trí.
- Giảm các yếu tố nhiễu.
- Làm rõ dữ liệu của **QR Code** và decode.

III. PHƯƠNG PHÁP TIẾP CẬN

Nghiên cứu các thuật toán xử lý ảnh thông dụng: Canny, threshold, median, gaussian, otsu,..

Nghiên cứu phương pháp có khả năng phát hiện và giải mã được **QR Code** thông dụng.

IV. PHƯƠNG PHÁP

1. Xử lý ảnh trước khi xác định vị trí.

- Chuyển ảnh đầu thành ảnh trắng đen:

```
cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

- Làm rõ các vùng trắng đen bằng **Otsu**:

```
cv2.threshold(gray, 120, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
```

- Giảm **Noise** bằng **Median** và **Gaussian**:

```
BLUR_VALUE = 3
```

```
cv2.medianBlur(gray, BLUR_VALUE)
```

```
cv2.GaussianBlur(gray, (BLUR_VALUE, BLUR_VALUE), 0)
```

- Xác định cạnh bằng **Canny**:

```
cv2.Canny(gray, 30, 200)
```

2. Xác định vị trí QR Code.

- Xác định các Contour:

```
cv2.findContours(edged.copy(), cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
```

- Tìm các ô vuông từ Contour.
- Tìm các ô vuông có kích thước tương tự.
- Xác định vị trí 4 ô vuông phần góc của **QR Code** nếu có.
- Xác định 4 góc của mã **QR Code**.
- Căn chỉnh và cắt để lấy được vùng **QR Code** hoàn chỉnh.

3. Xử lí ảnh QR Code trước khi decode.

- Làm nổi bật trắng đen lần nữa với **Threshold binary**:

```
cv2.threshold(warpCode, 100, 255, cv2.THRESH_BINARY)
```

- Loại bỏ các chấm li ti và làm rõ các vùng dữ liệu trên **QR Code**:

```
kernel = cv2.getStructuringElement(cv2.MORPH_RECT,(5,5))

warpCode1 = cv2.morphologyEx(warpCode0, cv2.MORPH_CLOSE, kernel)

warpCode1 = cv2.morphologyEx(warpCode1, cv2.MORPH_OPEN, kernel)
```

4. Decode.

Decode **QR Code** bằng thư viện **pyzbar** ở python.

```
decodedObjects = pyzbar.decode(warpCode)
```

- **TH1:** decode **QR Code** sau khi xử lí ở **mục 3**.
- **TH2:** nếu **TH1** không thể decode thì decode ảnh **QR Code** sau khi xử lí **mục 2**.

TÀI LIỆU THAM KHẢO

- Liu, Yue, Ju Yang, and Mingjun Liu. "Recognition of QR Code with mobile phones." *2008 Chinese control and decision conference*. IEEE, 2008.